In [1]:

```
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn.metrics import confusion_matrix
```

In [2]:

```
wine = datasets.load_wine()

```

```
In [3]: wine
```

```
Out[3]: {'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
                 1.065e+03],
                [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
                 1.050e+03],
                [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
                 1.185e+03],
                ...,
                [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
                 8.350e+02],
                [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
                 8.400e+02],
                [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
                 5.600e+02]]),
         'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
                2, 2]),
         'frame': None,
         'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
         'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\n------------------------\n\n**Data Set Charact
        eristics:**\n\n    :Number of Instances: 178 (50 in each of three classes)\n    :Number of Attributes: 1
        3 numeric, predictive attributes and the class\n    :Attribute Information:\n \t\t- Alcohol\n \t\t- Mali
        c acid\n \t\t- Ash\n\t\t- Alcalinity of ash  \n \t\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids
        \n \t\t- Nonflavanoid phenols\n \t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD280/O
        D315 of diluted wines\n \t\t- Proline\n\n    - class:\n            - class_0\n            - class_1\n
        - class_2\n\t\t\n    :Summary Statistics:\n    \n    ============================= ==== ===== ======= ==
        ===\n                                   Min   Max   Mean     SD\n    ============================= ====
        ===== ======= =====\n    Alcohol:                          11.0  14.8    13.0   0.8\n    Malic Acid:
        0.74  5.80    2.34  1.12\n    Ash:                              1.36  3.23    2.36  0.27\n    Alcalinity of
        Ash:            10.6  30.0    19.5   3.3\n    Magnesium:                        70.0 162.0    99.7  14.3\n
        Total Phenols:                0.98  3.88    2.29  0.63\n    Flavanoids:                       0.34  5.08
        2.03  1.00\n    Nonflavanoid Phenols:         0.13  0.66    0.36  0.12\n    Proanthocyanins:
        0.41  3.58    1.59  0.57\n    Colour Intensity:              1.3  13.0     5.1   2.3\n    Hue:
        0.48  1.71    0.96  0.23\n    OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71\n    Proline:
        278  1680     746   315\n    ============================= ==== ===== ======= =====\n\n    :Missing Attr
        ibute Values: None\n    :Class Distribution: class_0 (59), class_1 (71), class_2 (48)\n    :Creator: R.
        A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThis is
        a copy of UCI ML Wine recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/w
        ine/wine.data\n\nThe data is the results of a chemical analysis of wines grown in the same\nregion in It
        aly by three different cultivators. There are thirteen different\nmeasurements taken for different const
        ituents found in the three types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVUS - \nAn Exten
        dible Package for Data Exploration, Classification and Correlation. \nInstitute of Pharmaceutical and Fo
        od Analysis and Technologies,\nVia Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichman, M. (201
        3). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/ml]. Irvine, CA: University of Califor
        nia,\nSchool of Information and Computer Science. \n\n.. topic:: References\n\n  (1) S. Aeberhard, D. Co
        omans and O. de Vel, \n  Comparison of Classifiers in High Dimensional Settings, \n  Tech. Rep. no. 92-0
        2, (1992), Dept. of Computer Science and Dept. of  \n  Mathematics and Statistics, James Cook University
        of North Queensland. \n  (Also submitted to Technometrics). \n\n  The data was used with many others for
        comparing various \n  classifiers. The classes are separable, though only RDA \n  has achieved 100% corr
        ect classification. \n  (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) \n  (All resu
        lts using the leave-one-out technique) \n\n  (2) S. Aeberhard, D. Coomans and O. de Vel, \n  "THE CLASSI
        FICATION PERFORMANCE OF RDA" \n  Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \n
        Mathematics and Statistics, James Cook University of North Queensland. \n  (Also submitted to Journal of
        Chemometrics).\n',
         'feature_names': ['alcohol',
          'malic_acid',
          'ash',
          'alcalinity_of_ash',
          'magnesium',
          'total_phenols',
          'flavanoids',
          'nonflavanoid_phenols',
          'proanthocyanins',
          'color_intensity',
          'hue',
          'od280/od315_of_diluted_wines',
          'proline']}
```

In [4]:

```
df = pd.DataFrame(wine['data'], columns=wine['feature_names'])
```

In [5]:

```
df
```

Out[5]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_int
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	
...	...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	

178 rows × 13 columns

In [6]:

```
df['target'] = wine['target']
```

In [7]:

```
df
```

Out[7]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_int
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	
...	...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	

178 rows × 14 columns

In [8]:

```
df.shape
```

Out[8]:

```
(178, 14)
```

In [9]:

```
df.isna().sum()
```

Out[9]:

```
alcohol                         0
malic_acid                      0
ash                             0
alcalinity_of_ash               0
magnesium                       0
total_phenols                   0
flavanoids                      0
nonflavanoid_phenols            0
proanthocyanins                 0
color_intensity                 0
hue                             0
od280/od315_of_diluted_wines    0
proline                         0
target                          0
dtype: int64
```

seperate our data

In [10]:

```
x = df.drop('target', axis = 1)
y = df['target']
```

In [11]:

```
x, y
```

```
Out[11]: (     alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
          0      14.23        1.71  2.43               15.6      127.0           2.80   
          1      13.20        1.78  2.14               11.2      100.0           2.65   
          2      13.16        2.36  2.67               18.6      101.0           2.80   
          3      14.37        1.95  2.50               16.8      113.0           3.85   
          4      13.24        2.59  2.87               21.0      118.0           2.80   
          ..       ...         ...   ...                ...        ...            ...   
          173    13.71        5.65  2.45               20.5       95.0           1.68   
          174    13.40        3.91  2.48               23.0      102.0           1.80   
          175    13.27        4.28  2.26               20.0      120.0           1.59   
          176    13.17        2.59  2.37               20.0      120.0           1.65   
          177    14.13        4.10  2.74               24.5       96.0           2.05   

               flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
          0          3.06                  0.28             2.29             5.64  1.04   
          1          2.76                  0.26             1.28             4.38  1.05   
          2          3.24                  0.30             2.81             5.68  1.03   
          3          3.49                  0.24             2.18             7.80  0.86   
          4          2.69                  0.39             1.82             4.32  1.04   
          ..          ...                   ...              ...              ...   ...   
          173        0.61                  0.52             1.06             7.70  0.64   
          174        0.75                  0.43             1.41             7.30  0.70   
          175        0.69                  0.43             1.35            10.20  0.59   
          176        0.68                  0.53             1.46             9.30  0.60   
          177        0.76                  0.56             1.35             9.20  0.61   

               od280/od315_of_diluted_wines  proline  
          0                            3.92   1065.0  
          1                            3.40   1050.0  
          2                            3.17   1185.0  
          3                            3.45   1480.0  
          4                            2.93    735.0  
          ..                            ...      ...  
          173                          1.74    740.0  
          174                          1.56    750.0  
          175                          1.56    835.0  
          176                          1.62    840.0  
          177                          1.60    560.0  

          [178 rows x 13 columns],
          0      0
          1      0
          2      0
          3      0
          4      0
                ..
          173    2
          174    2
          175    2
          176    2
          177    2
          Name: target, Length: 178, dtype: int32)
```

In [12]:

```python
x.shape, y.shape
```

Out[12]:

```
((178, 13), (178,))
```

Train, Test split

In [13]:

```python
from sklearn.model_selection import train_test_split
```

In [14]:

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y, random_state=42)
```

In [15]:

```python
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[15]:

```
((142, 13), (36, 13), (142,), (36,))
```

Train our model

In [16]:

```python
from sklearn.neighbors import KNeighborsClassifier
```

In [17]:

```python
# Create an instance of the KNeighborsClassifier with n_neighbors = 5
knn = KNeighborsClassifier(n_neighbors=10)

```

In [18]:

```python
knn.fit(x_train, y_train)
```

Out[18]:

```
KNeighborsClassifier(n_neighbors=10)
```

In [19]:

```
knn.score(x_test, y_test)
```

```
C:\Users\USER\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unlik
e other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserve
s the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` wi
ll become False, the `axis` over which the statistic is taken will be eliminated, and the value None wil
l no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

Out[19]:

```
0.8333333333333334
```

Turnig model sensitivity

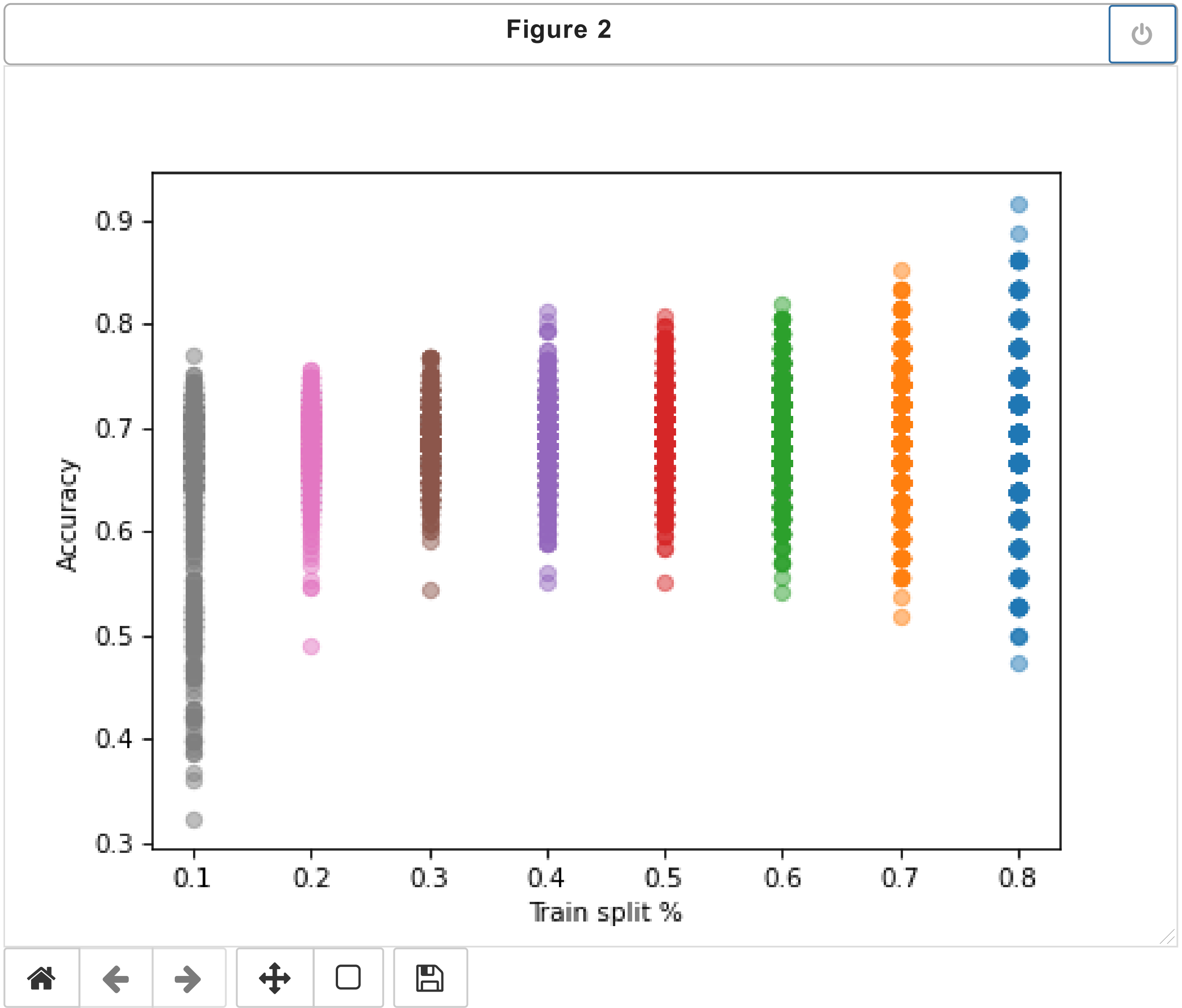
In [20]:

```python
k_range = range(1, 25)
scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train, y_train)
    scores.append(knn.score(x_test, y_test))
    import warnings
warnings.filterwarnings('ignore')

#makes the plot interactive
%matplotlib notebook
plt.figure()
plt.xlabel('k count')
plt.ylabel('Model Accuracy')
plt.scatter(k_range, scores)
plt.grid()
plt.xticks([0, 5, 10, 15, 20, 30])
plt.show();
```

```
C:\Users\USER\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unl
ike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically pres
erves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdi
ms` will become False, the `axis` over which the statistic is taken will be eliminated, and the value
None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unl
ike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically pres
erves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdi
ms` will become False, the `axis` over which the statistic is taken will be eliminated, and the value
None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unl
ike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically pres
erves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdi
ms` will become False, the `axis` over which the statistic is taken will be eliminated, and the value
None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:228: FutureWarning: Unl
```

In [21]:

```python

test_sizes = [0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]
knn = KNeighborsClassifier(n_neighbors=5)

plt.figure()

for test_size in test_sizes:
    scores = []

    for i in range(1, 1000):
        x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=1 - test_size)
        knn.fit(x_train, y_train)
        scores.append(knn.score(x_test, y_test))

    plt.plot(test_size, np.mean(scores))
    plt.scatter([test_size] * len(scores), scores, alpha=0.5)  # Scatter plot

plt.xlabel('Train split %')
plt.ylabel('Accuracy')
plt.show()
```



Make predictions

In [22]:

```
prediction = knn.predict(x_test)
```

```
In [23]: prediction
```

```
Out[23]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0,
                1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0,
                1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1,
                1, 0, 1, 1, 1, 1, 0])
```

```
In [24]: y
```

```
Out[24]: 0      0
         1      0
         2      0
         3      0
         4      0
               ..
         173    2
         174    2
         175    2
         176    2
         177    2
         Name: target, Length: 178, dtype: int32
```

```
In [25]: y_test
```

```
Out[25]: 162    2
         140    2
         177    2
         176    2
         115    1
               ..
         61     1
         77     1
         43     0
         108    1
         37     0
         Name: target, Length: 161, dtype: int32
```

In [26]:

```
# Generate the confusion matrix
cm = confusion_matrix(y_test, prediction)
cm
```

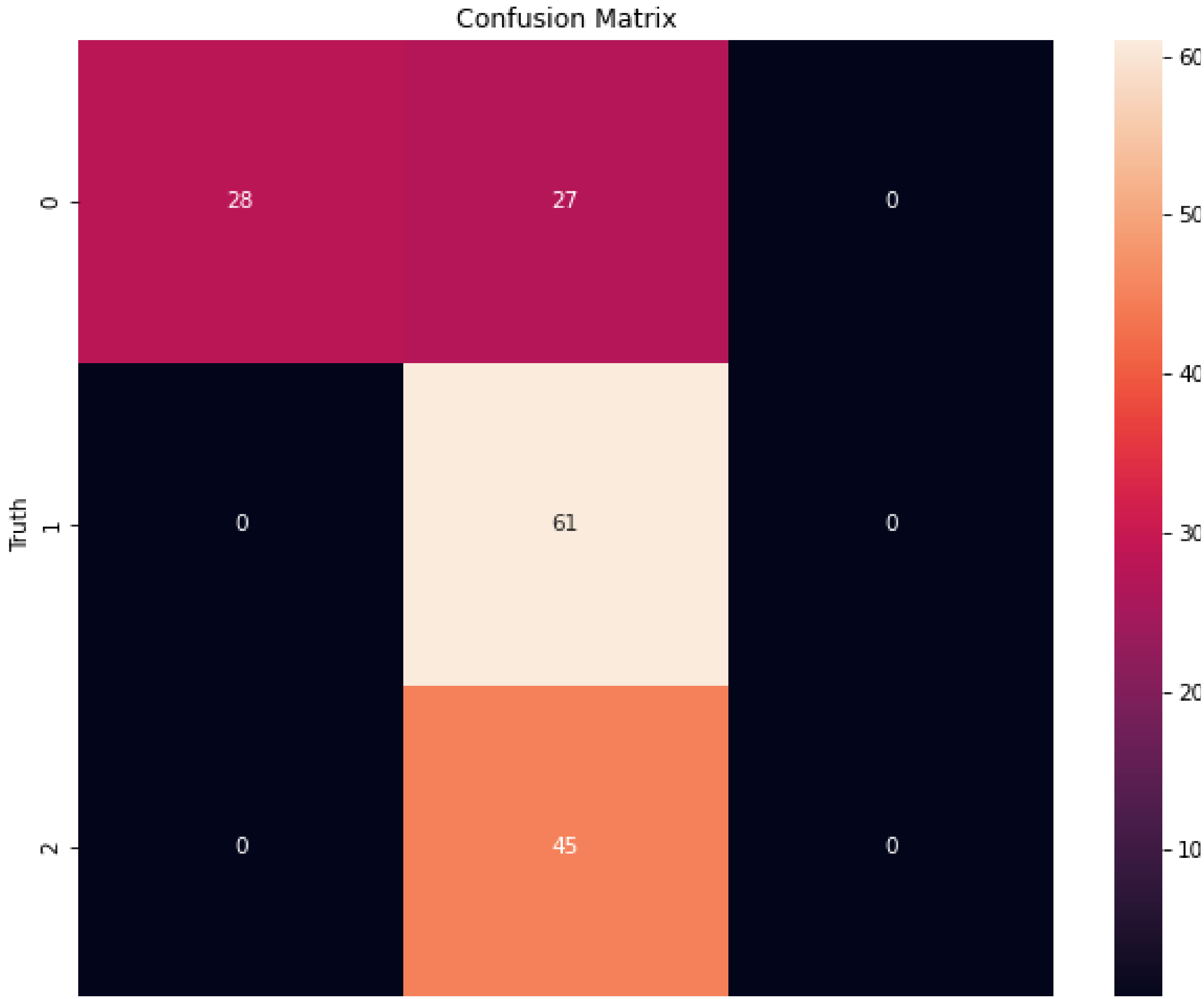
Out[26]:

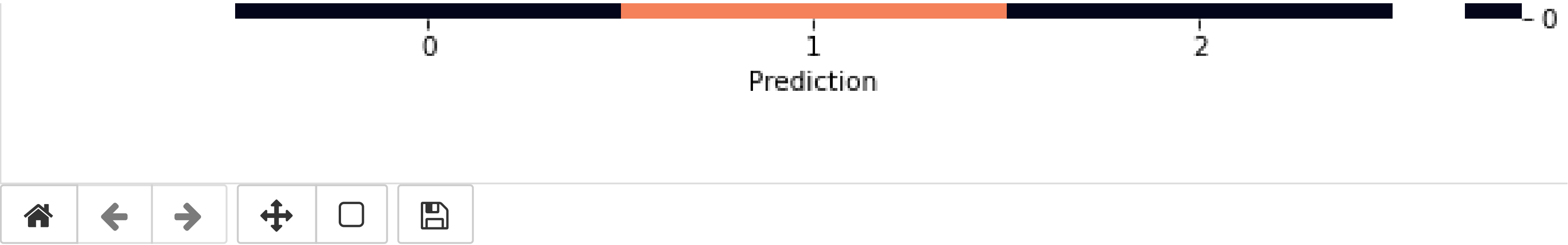
```
array([[28, 27,  0],
       [ 0, 61,  0],
       [ 0, 45,  0]], dtype=int64)
```

In [27]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming cm contains the confusion matrix generated previously

plt.figure(figsize=(10, 8))  # Adjust the figure size as needed
sns.heatmap(cm, annot=True)
plt.title('Confusion Matrix')
plt.ylabel('Truth')
plt.xlabel('Prediction')
plt.show()
```

Figure 3





In [ ]: